METHOD, SYSTEM, AND COMPUTER PROGRAM PRODUCT FOR ANALYZING COMBINATORIAL LIBRARIES

Inventors: Dimitris K. Agrafiotis
Victor S. Lobanov
F. Raymond Salemme

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation-in-part of U.S. Application No. 09/934,084, filed August 22, 2001, which is incorporated by reference herein in its entirety, and it claims the benefit of U.S. Provisional Application No. 60/264,258, filed January 29, 2001, and U.S. Provisional Application No. 60/274,238, filed March 9, 2001, each of which is incorporated by reference herein in its entirety.

FIELD OF THE INVENTION

[0002] The present invention relates to combinatorial chemistry and computer aided molecular design. The present invention also relates to pattern analysis, information representation, information cartography and data mining. In particular, the present invention relates to generating mapping coordinates for products in a combinatorial chemical library based on reagent data.

BACKGROUND OF THE INVENTION

[0003] Molecular similarity is one of the most ubiquitous concepts in chemistry (Johnson, M. A., and Maggiora, G. M., *Concepts and Applications of Molecular Similarity*. Wiley, New York (1990)). It is used to analyze and categorize chemical phenomena, rationalize the behavior and function of molecules, and design new chemical entities with improved physical, chemical, and biological properties. Molecular similarity is typically quantified in the form of a numerical index derived either through direct observation, or through the measurement of a set of characteristic properties (descriptors), which are subsequently combined in some form of dissimilarity

or distance measure. For large collections of compounds, similarities are usually described in the form of a symmetric matrix that contains all the pairwise relationships between the molecules in the collection. Unfortunately, pairwise similarity matrices do not lend themselves for numerical processing and visual inspection. A common solution to this problem is to embed the objects into a low-dimensional Euclidean space in a way that preserves the original pairwise proximities as faithfully as possible. This approach, known as multidimensional scaling (MDS) (Torgeson, W. S., *Psychometrika 17*:401-419 (1952); Kruskal, J. B., *Phychometrika 29*:115-129 (1964)) or nonlinear mapping (NLM) (Sammon, J. W., *IEEE Trans. Comp. C18*:401-409 (1969)), converts the data points into a set of real-valued vectors that can subsequently be used for a variety of pattern recognition and classification tasks.

[0004] Given a set of k objects, a symmetric matrix, r_{ij}, of relationships between these objects, and a set of images on a m-dimensional map $\{y_i, i = 1, 2, .., k; y_i \in \Re^m\}$, the problem is to place y_i onto the map in such a way that their Euclidean distances $d_{ij} = \|y_i - y_j\|$ approximate as closely as possible the corresponding values r_{ij}. The quality of the projection is determined using a sum-of-squares error function known as stress, which measures the differences between d_{ij} and r_{ij} over all $k(k\text{-}1)/2$ possible pairs. This function is numerically minimized in order to generate the optimal map. This is typically carried out in an iterative fashion by: (1) generating an initial set of coordinates y_i, (2) computing the distances d_{ij}, (3) finding a new set of coordinates y_i that lead to a reduction in stress using a steepest descent algorithm, and (4) repeating steps (2) and (3) until the change in the stress function falls below some predefined threshold. There is a wide variety of MDS algorithms involving different error (stress) functions and optimization heuristics, which are reviewed in Schiffman, Reynolds and Young, *Introduction to Multidimensional Scaling*, Academic Press, New York (1981); Young and Hamer, *Multidimensional Scaling: History, Theory and Applications*, Erlbaum Associates, Inc., Hillsdale, NJ (1987); Cox and Cox, *Multidimensional Scaling*, Number 59 in *Monographs in Statistics and Applied Probability*, Chapman-Hall (1994), and Borg, I., Groenen, P., Modern Multidimensional Scaling, Springer-Verlag,

New York, (1997). The contents of these publications are incorporated herein by reference in their entireties.

[0005] Unfortunately, the quadratic nature of the stress function (i.e. the fact that the computational time required scales proportionally to k^2) make these algorithms impractical for data sets containing more than a few thousand items. Several attempts have been devised to reduce the complexity of the task. (See Chang, C. L., and Lee, R. C. T., *IEEE Trans. Syst., Man, Cybern.*, 1973, *SMC-3*, 197-200; Pykett, C. E., *Electron. Lett.*, 1978, *14*, 799-800; Lee, R. C. Y., Slagle, J. R., and Blum, H., *IEEE Trans. Comput.*, 1977, *C-27*, 288-292; Biswas, G., Jain, A. K., and Dubes, R. C., *IEEE Trans. Pattern Anal. Machine Intell.*, 1981, *PAMI-3(6)*, 701-708). However, these methods either focus on a small subset of objects or a small fraction of distances, and the resulting maps are generally difficult to interpret.

[0006] Recently, two very effective alternative strategies were described. The first is based on a self-organizing procedure which repeatedly selects subsets of objects from the set of objects to be mapped, and refines their coordinates so that their distances on the map approximate more closely their corresponding relationships. (U.S. Patent No. 6,295,514, and U.S. Application No. 09/073,845, filed May 7, 1998, each of which is incorporated by reference herein in its entirety). The method involves the following steps: (1) placing the objects on the map at some initial coordinates, y_i, (2) selecting a subset of objects, (3) revising the coordinates, y_i, of at least some of the selected objects so that at least some of their distances, d_{ij}, match more closely their corresponding relationships r_{ij}, (4) repeating steps (2) and (3) for additional subsets of objects, and (4) exporting the refined coordinates, y_i, for the entire set of objects or any subset thereof.

[0007] The second method attempts to derive an analytical mapping function that can generate mapping coordinates from a set of object features. (See U.S. Application No. 09/303,671, filed May 3, 1999, and U.S. Application No. 09/814,160, filed March 22, 2001, each of which is incorporated by reference herein in its entirety). The method works as follows. Initially, a subset of objects from the set of objects to be mapped and their associated relationships are selected. This subset of objects is then mapped onto an m-dimensional map

using the self-organizing procedure described above, or any other MDS algorithm. Hereafter, the coordinates of objects in this m-dimensional map shall be referred to as "output coordinates" or "output features". In addition, a set of n attributes are determined for each of the selected subset of objects. Hereafter, these n attributes shall be referred to as "input coordinates" or "input features". Thus, each object in the selected subset of objects is associated with an n-dimensional vector of input features and an m-dimensional vector of output features. A supervised machine learning approach is then employed to determine a functional relationship between the n-dimensional input and m-dimensional output vectors, and that functional relationship is recorded. Hereafter, this functional relationship shall be referred to as a "mapping function". Additional objects that are not part of the selected subset of objects may be mapped by computing their input features and using them as input to the mapping function, which produces their output coordinates. The mapping function can be encoded in a neural network or a collection of neural networks.

[0008] Both the self-organizing and the neural network methods are general and can be used to produce maps of any desired dimensionality.

[0009] MDS can be particularly valuable for analyzing and visualizing combinatorial chemical libraries. A combinatorial library is a collection of chemical compounds derived from the systematic combination of a prescribed set of chemical building blocks according to a specific reaction protocol. A combinatorial library is typically represented as a list of variation sites on a molecular scaffold, each of which is associated with a list of chemical building blocks. Each compound (or product) in a combinatorial library can be represented by a unique tuple, $\{r_1, r_2, \ldots, r_d\}$, where r_i is the building block at the i-th variation site, and d is the number of variation sites in the library. For example, a polypeptide combinatorial library is formed by combining a set of chemical building blocks called amino acids in every possible way for a given compound length (here, the number of variation sites is the number of amino acids along the polypeptide chain). Millions of products theoretically can be synthesized through such combinatorial mixing of building blocks. As one commentator has observed, the systematic combinatorial mixing of 100

interchangeable chemical building blocks results in the theoretical synthesis of 100 million tetrameric compounds or 10 billion pentameric compounds (Gallop *et al.*, "Applications of Combinatorial Technologies to Drug Discovery, Background and Peptide Combinatorial Libraries," J. Med. Chem. 37, 1233-1250 (1994), which is incorporated by reference herein in its entirety). A computer representation of a combinatorial library is often referred to as a virtual combinatorial library.

[0010] MDS can simplify the analysis of combinatorial libraries in two important ways: (1) by reducing the number of dimensions that are required to describe the compounds in some abstract chemical property space in a way that preserves the original relationships among the compounds, and (2) by producing Cartesian coordinate vectors from data supplied directly or indirectly in the form of molecular similarities, so that they can be analyzed with conventional statistical and data mining techniques. Typical applications of coordinates obtained with MDS include visualization, diversity analysis, similarity searching, compound classification, structure-activity correlation, etc. (See, e.g., Agrafiotis, D. K., The diversity of chemical libraries, *The Encyclopedia of Computational Chemistry*, Schleyer, P. v. R., Allinger, N. L., Clark, T., Gasteiger, J., Kollman, P. A., Schaefer III, H. F., and Schreiner, P. R., Eds., John Wiley & Sons, Chichester, 742-761 (1998); and Agrafiotis, D. K., Myslik, J. C., and Salemme, F. R., Advances in diversity profiling and combinatorial series design, *Mol. Diversity, 4(1)*, 1-22 (1999), each of which is incorporated by reference herein in its entirety).

[0011] Analyzing a combinatorial library based on the properties of the products (as opposed to the properties of their building blocks) is often referred to as product-based design. Several product-based methodologies for analyzing virtual combinatorial libraries have been developed. (See, e.g., Sheridan, R.P., and Kearsley, S.K., Using a genetic algorithm to suggest combinatorial libraries, *J. Chem. Info. Comput. Sci., 35*, 310-320 (1995); Weber, L., Wallbaum, S., Broger, C., and Gubernator, K., Optimization of the biological activity of combinatorial compound libraries by a genetic algorithm, *Angew. Chem. Int. Ed. Eng., 34*, 2280-2282 (1995); Singh, J., Ator, M. A., Jaeger, E. P., Allen, M. P., Whipple, D. A., Soloweij, J. E., Chowdhary, S.,

and Treasurywala, A. M., Application of genetic algorithms to combinatorial synthesis: a computational approach for lead identification and lead optimization, *J. Am. Chem. Soc.*, *118*, 1669-1676 (1996); Agrafiotis, D. K., Stochastic algorithms for maximizing molecular diversity, *J. Chem. Info. Comput. Sci.*, *37*, 841-851 (1997); Brown, R. D., and Martin, Y. C., Designing combinatorial library mixtures using genetic algorithms, *J. Med. Chem.*, *40*, 2304-2313 (1997); Murray, C.W., Clark, D.E., Auton, T.R., Firth, M.A., Li, J., Sykes, R.A., Waszkowycz, B., Westhead, D.R. and Young, S.C., PRO_SELECT: combining structure-based drug design and combinatorial chemistry for rapid lead discovery. 1. Technology, *J. Comput.-Aided Mol. Des.*, 11, 193-207 (1997); Agrafiotis, D. K., and Lobanov, V. S., An efficient implementation of distance-based diversity metrics based on k-d trees, *J. Chem. Inf. Comput. Sci.*, *39*, 51-58 (1999); Gillett, V. J., Willett, P., Bradshaw, J., and Green, D. V. S., Selecting combinatorial libraries to optimize diversity and physical properties, *J. Chem. Info. Comput. Sci.*, *39*, 169-177 (1999); Stanton, R. V., Mount, J., and Miller, J. L., Combinatorial library design: maximizing model-fitting compounds with matrix synthesis constraints, *J. Chem. Info. Comput. Sci.*, *40*, 701-705 (2000); and Agrafiotis, D. K., and Lobanov, V. S., Ultrafast algorithm for designing focused combinatorial arrays, *J. Chem. Info. Comput. Sci.*, *40*, 1030-1038 (2000), each of which is incorporated by reference herein in its entirety).

[0012] However, as will be understood by a person skilled in the relevant art(s), this approach requires explicit enumeration (i.e., virtual synthesis) of the products in the virtual library. This process can be prohibitively expensive when the library contains a large number of products. That is, the analysis cannot be accomplished in a reasonable amount of time using available computing systems. In such cases, the most common solution is to restrict attention to a smaller subset of products from the virtual library, or to consider each variation site independently of all the others. (See, e.g., Martin, E. J., Blaney, J. M., Siani, M. A., Spellmeyer, D. C., Wong, A. K., and Moos, W. H., *J. Med. Chem.*, *38*, 1431-1436 (1995); Martin, E. J., Spellmeyer, D. C., Critchlow, R. E. Jr., and Blaney, J. M., *Reviews in Computational Chemistry*, Vol. 10, Lipkowitz, K. B., and Boyd, D. B., Eds., VCH, Weinheim (1997);

and Martin, E., and Wong, A., Sensitivity analysis and other improvements to tailored combinatorial library design, *J. Chem. Info. Comput. Sci.*, *40*, 215-220 (2000), each of which is incorporated by reference herein in its entirety). Unfortunately, the latter approach, which is referred to as reagent-based design, often produces inferior results. (See, e.g., Gillet, V. J., Willett, P., and Bradshaw, J., *J. Chem. Inf. Comput. Sci.*; *37(4)*, 731-740 (1997); and Jamois, E. A., Hassan, M., and Waldman, M., Evaluation of reagent-based and product-based strategies in the design of combinatorial library subsets, *J. Chem. Inf. Comput. Sci.*, *40*, 63-70 (2000), each of which is incorporated by reference herein in its entirety).

[0013] Hence there is a need for methods, systems, and computer program products that can be used to analyze large combinatorial chemical libraries, which do not have the limitations discussed above. In particular, there is a need for methods, systems, and computer program products for rapidly generating mapping coordinates for compounds in a combinatorial library that do not require the enumeration of every possible product in the library.

SUMMARY OF THE INVENTION

[0014] The present invention provides a method, system, and computer program product for generating mapping coordinates of combinatorial library products from features of library building blocks.

[0015] As described herein, at least one feature is determined for each building block of a combinatorial library having a plurality of products. A training subset of products is selected from the plurality of products in the combinatorial library, and at least one mapping coordinate is determined for each product in the training subset of products. A set of building blocks is identified for each product in the training subset of products, and features associated with these building blocks are combined to form an input features vector for each product in the training subset of products. A supervised machine learning approach is used to infer a mapping function f that transforms the input features vector to the corresponding at least one mapping coordinate for each product in the training subset of products. The mapping

function f is encoded in a computer readable medium. After the mapping function f is inferred, it is used for determining, estimating, or generating mapping coordinates of other products in the combinatorial library. Mapping coordinates of other products are determined, estimated, or generated from their corresponding input features vectors using the inferred mapping function f. Sets of building blocks are identified for a plurality of additional products in the combinatorial library. Input features vectors are formed for the plurality of additional products. The input features vectors for the plurality of additional products are transformed using the mapping function f to obtain at least one estimated mapping coordinate for each of the plurality of additional products.

[0016] In embodiments of the invention, laboratory-measured values and/or computed values are used as features for the building blocks of the combinatorial library. In embodiments of the invention, at least one of the features of the building blocks at a particular variation site in the combinatorial library is the same as at least one of the features of the building blocks at a different variation site in the library. In accordance with the invention, features of building blocks represent reagents used to construct the combinatorial library, fragments of reagents used to construct the combinatorial library, and/or modified fragments of reagents used to construct the combinatorial library. Other features that can be used in accordance with the invention will become apparent to individuals skilled in the relevant arts given the description of the invention herein.

[0017] In an embodiment, the mapping function f is implemented using a neural network. The neural network is trained to implement the mapping function using the input features vector and the corresponding at least one mapping coordinate for each product of the training subset of products.

[0018] In other embodiments, the mapping function f is a set of specialized mapping functions f_1 through f_n. In an embodiment, each such specialized mapping function is implemented using a neural network.

[0019] In an embodiment, the mapping coordinates for the training subset of products are obtained by generating an initial set of mapping coordinates for the training subset of products and refining the coordinates in an iterative manner. In an embodiment, this is accomplished by selecting two products

from the training subset of products and refining the mapping coordinates of at least one of the selected products based on the coordinates of the two products and a distance between the two products. The mapping coordinates of at least one of the selected products are refined so that the distance between the refined coordinates of the two products is more representative of a relationship between the products. This process is typically repeated for additional products of the training subset of products until a stop criterion is satisfied.

[0020] In another embodiment, the generation of mapping coordinates for the training subset of products is accomplished by selecting at least three products from the training subset of products and refining the mapping coordinates of at least some of the selected products based on the coordinates of at least some of the selected products and at least some of the distances between the selected products. The mapping coordinates of at least some of the selected products are refined so that at least some of the distances between the refined coordinates of at least some of the selected products are more representative of corresponding relationships between the products. This process is typically repeated for additional subsets of products from the training subset of products until a stop criterion is satisfied.

[0021] In other embodiments, the mapping coordinates for the training subset of products are generated using multidimensional scaling or nonlinear mapping so that the distances between the mapping coordinates of the products in the training subset of products are representative of corresponding relationships between the products.

[0022] In other embodiments, the mapping coordinates for the training subset of products are obtained from a different mapping function f^*. In one embodiment, the mapping function f^* takes as input a set of features associated with each product in the training subset of products and produces the corresponding at least one mapping coordinate for each product in the training subset of products. In another embodiment, the mapping function f^* takes as input a set of features associated with building blocks associated with each product in the training subset of products and produces the corresponding at least one mapping coordinate for each product in the training subset of products.

[0023] In other embodiments, the mapping coordinates for the training subset of products are obtained from a computer readable medium.

[0024] Further embodiments, features, and advantages of the present invention, as well as the structure and operation of the various embodiments of the present invention, are described in detail below with reference to the accompanying figures.

BRIEF DESCRIPTION OF THE DRAWINGS/FIGURES

[0025] The present invention is described with reference to the accompanying drawings wherein:

[0026] FIG.1 illustrates an example combinatorial neural network according to an embodiment of the present invention;

[0027] FIGs. 2A-B illustrate a flowchart of a method for generating coordinates for products in a combinatorial library according to an embodiment of the present invention;

[0028] FIGs. 3A-B illustrate a flowchart of a second method for generating coordinates for products in a combinatorial library according to an embodiment of the present invention;

[0029] FIG. 4 illustrates a reaction scheme for a reductive amination combinatorial library;

[0030] FIG. 5A illustrates an example two-dimensional nonlinear map for a reductive amination library using Kier-Hall descriptors obtained by a non-linear mapping method;

[0031] FIG. 5B illustrates an example two-dimensional nonlinear map for a reductive amination library using Kier-Hall descriptors obtained by a combinatorial neural network according to the present invention;

[0032] FIG. 6A illustrates an example two-dimensional nonlinear map for a reductive amination library using Isis Keys descriptors obtained by a non-linear mapping method;

[0033] FIG. 6B illustrates an example two-dimensional nonlinear map for a reductive amination library using Isis Keys descriptors obtained by a combinatorial neural network according to the present invention; and

[0034] FIG. 7 illustrates an exemplary computing environment within which the invention can operate.

DETAILED DESCRIPTION OF THE INVENTION

[0035] Preferred embodiments of the present invention are now described with references to the figures, where like reference numbers indicate identical or functionally similar elements. Also in the figures, the left most digit(s) of each reference number corresponds to the figure in which the reference number is first used. While specific configurations and arrangements are discussed, it should be understood that this is done for illustrative purposes only. One skilled in the relevant art will recognize that other configurations and arrangements can be used without departing from the spirit and scope of the invention. It will also be apparent to one skilled in the relevant art(s) that this invention can also be employed in a variety of other devices and applications, and is not limited to just the embodiments described herein.

1. Overview of the Invention

[0036] The present invention provides a method, system, and computer program product for generating mapping coordinates of combinatorial library products from features of library building blocks. In operation, features of library building blocks and mapping coordinates for a training subset of products in the combinatorial library are obtained and used to infer a mapping function *f* that transforms building block features to mapping coordinates for each product in the training subset of products. The mapping function *f* is encoded in a computer readable medium.

[0037] The mapping function *f* can be retrieved and used to generate mapping coordinates for additional products in the combinatorial library from features of building blocks associated with the additional products. In an embodiment, after the mapping function *f* is inferred, mapping coordinates of additional products in the combinatorial library are generated by obtaining features of the building blocks and using them as input to the mapping function *f*, which

generates mapping coordinates for the additional library products. The mapping coordinates can then be used for any subsequent analysis, searching, or classification. As will be understood by a person skilled in the relevant art, given the description herein, the present invention can be applied to a wide variety of mapping coordinates and/or building block features.

2. Combinatorial Neural Networks

[0038] As described below, in some embodiments of the invention the mapping function f is implemented using a neural network. This neural network is hereafter referred to as a combinatorial network or combinatorial neural network. The combinatorial network is trained to generate at least one mapping coordinate of the combinatorial library products from input features of their respective building blocks. As used herein, the term "mapping coordinates" refers to the mapping coordinates of the library products, and the term "building block features" refers to the input features of the library building blocks (e.g., reagents, fragments of reagents, and/or modified fragments of reagents).

[0039] Generally speaking, a combinatorial network comprises an input layer containing $n_1+n_2+\ldots+n_r$ neurons, where r is the number of variation sites in the combinatorial library and n_i is the number of input features of the building blocks at the i-th variation site. In addition, a combinatorial network comprises one or more hidden layers containing one or more neurons each, and an output layer having a single neuron for each mapping coordinate generated by the neural network.

[0040] FIG. 1 illustrates an example combinatorial network 100 according to an embodiment of the invention. Combinatorial network 100 is a fully connected multilayer perceptron (MLP). In accordance with the invention, the outputs of combinatorial network 100 represent mapping coordinates of the library products.

[0041] As illustrated in FIG. 1, combinatorial network 100 has an input layer 102, a hidden layer 104, and an output layer 106. In an embodiment, a nonlinear transfer function, such as the logistic transfer function

$f(x)=1/(1+e^{-x})$, is used for the hidden and/or output layers. Combinatorial network 100 can be trained in accordance with the invention using, for example, the error back-propagation algorithm (see, e.g., S. Haykin, Neural Networks, Macmillan, New York (1994), which is incorporated by reference herein in its entirety). Other neural network architectures and/or training algorithms that can be used in accordance with the invention will become apparent to individuals skilled in the relevant arts given the description of the invention herein.

[0042] As will be understood by persons skilled in the relevant art given the description herein, training data used to train a combinatorial network typically include two sets of parameters. The first set consists of one or more input features for each of the library building blocks. The second set consists of one or more mapping coordinates for the training subset of products. The building block features are concatenated into a single array, and are presented to the network in the same order (e.g., $f_{11}, f_{12}, \ldots, f_{1n1}, f_{21}, f_{22}, \ldots, f_{2n2}, \ldots, f_{r1}, f_{r2}, \ldots, f_{rnr}$, where f_{ij} is the j-th feature of the building block at the i-th variation site).

[0043] In an embodiment, the training subset of products presented to a network is determined by random sampling. (See Agrafiotis, D. K., and Lobanov, V. S., Nonlinear Mapping Networks. *J. Chem. Info. Comput. Sci.*, 40, 1356-1362 (2000), which is incorporated by reference herein in its entirety).

3. Method Embodiments of the Invention

[0044] As described herein, the invention permits the *in silico* characterization and analysis of large virtual combinatorial libraries. A virtual combinatorial library is an electronic representation of a collection of chemical compounds or "products" generated by the systematic combination of a number of chemical "building blocks" such as reagents according to a specific reaction protocol. Typically, embodiments of the invention are significantly faster than

conventional library analysis methodologies that are based on full enumeration of the combinatorial products.

A. Example Method 200

[0045] FIGs. 2A and 2B illustrate a flowchart of the steps of a method 200 for generating mapping coordinates of products in a virtual combinatorial library based on features of corresponding building blocks. Typically, distances between the mapping coordinates of products represent relationships between the products.

[0046] The steps of method 200 will now be described with reference to FIGs. 2A and 2B. Method 200 begins with step 202.

[0047] In step 202, mapping coordinates are obtained for a training subset of products in the virtual combinatorial library.

[0048] In an embodiment of the invention, a training subset of products from the virtual combinatorial library is identified in step 202. Relationships between products in the training subset of products are then obtained and are used to produce mapping coordinates for the products in the training subset of products.

[0049] In an embodiment, distances between mapping coordinates of the products in the training subset of products are representative of corresponding relationships between the products.

[0050] In other embodiments, mapping coordinates for the products in the training subset of products are obtained in step 202 by generating an initial set of mapping coordinates for the products in the training subset of products and refining the coordinates in an iterative manner until a stop criterion is satisfied. This may be accomplished, for example, by selecting two products at a time from the training subset of products and refining the mapping coordinates of at least one of the selected products based on the coordinates of the two products and a distance between the two products. The mapping coordinates of at least one of the selected products is refined so that the distance between the refined coordinates of the two products is more representative of a relationship

between the products. This mapping process is further described in U.S. Patent No. 6,295,514, and U.S. Application No. 09/073,845, filed May 7, 1998.

[0051] In other embodiments, the generation of mapping coordinates for the products in the training subset of products is accomplished by selecting at least three products from the training subset of products and refining the mapping coordinates of at least some of the selected products based on the coordinates of at least some of the selected products and at least some of the distances between the selected products. The mapping coordinates of at least some of the selected products are refined so that at least some of the distances between the refined coordinates of at least some of the selected products are more representative of corresponding relationships between the products. This process is typically repeated for additional subsets of products from the training subset of products until a stop criterion is satisfied. This mapping process is further described in U.S. Patent No. 6,295,514, and U.S. Application No. 09/073,845, filed May 7, 1998.

[0052] In other embodiments, the mapping coordinates for the training subset of products are generated using multidimensional scaling or nonlinear mapping so that the distances between the mapping coordinates of the products in the training subset of products are representative of corresponding relationships between the products.

[0053] In other embodiments, the mapping coordinates for the training subset of products are obtained from a different mapping function f^*. In one embodiment, the mapping function f^* takes as input a set of features associated with each product in the training subset of products and produces the corresponding at least one mapping coordinate for each product in the training subset of products. In another embodiment, the mapping function f^* takes as input a set of features associated with building blocks associated with each product in the training subset of products and produces the corresponding at least one mapping coordinate for each product in the training subset of products.

[0054] In an embodiment, relationships between products in the training subset of products are obtained by obtaining a set of properties for each product in the training subset of products, and computing relationships

between products using the properties of the training subset of products. As will be understood by persons skilled in the relevant art, any relationship measure that can relate products in the training subset of products can be used in this regard. In an embodiment, relationships between products represent similarities or dissimilarities between the products.

[0055] In other embodiments, mapping coordinates for the products in the training subset of products are obtained by obtaining a set of properties for each product in the training subset of products, and computing a set of latent coordinates from at least some of the properties of the training subset of products using a dimensionality reduction method.

[0056] In other embodiments, the mapping coordinates for the training subset of products are obtained in step 202, for example, by retrieving the mapping coordinates from a computer readable medium.

[0057] Other means that can be used in accordance with the invention to obtain mapping coordinates for the training subset of products will become apparent to individuals skilled in the relevant arts given the description of the invention herein.

[0058] In step 204, building block features (i.e., numerical representations of the building blocks of the combinatorial library) are obtained for the training subset of products. These building block features can be obtained in any desired manner. Furthermore, there is no requirement in step 204 to obtain the same type of numerical representations for the library building blocks as those obtained in step 202 for the training subset of products.

[0059] In embodiments of the invention, laboratory-measured values and/or computed values are used as features for the building blocks of the combinatorial library. In embodiments of the invention, at least one of the features of the building blocks at a particular variation site in the combinatorial library is the same as at least one of the features of the building blocks at a different variation site in the library.

[0060] In accordance with the invention, features of building blocks represent reagents used to construct the combinatorial library, fragments of reagents used to construct the combinatorial library, and/or modified fragments of reagents used to construct the combinatorial library. Other features that can be

used in accordance with the invention will become apparent to individuals skilled in the relevant arts given the description of the invention herein.

[0061] In step 206, a supervised machine learning approach is used to infer a mapping function *f* that transforms the building block features for each product in the training subset of products to the corresponding mapping coordinates for each product in the training subset of products. In embodiments of the invention, step 206 involves training a combinatorial neural network to transform the building block features for each product in the training subset of products to the corresponding mapping coordinates for each product in the training subset of products.

[0062] In step 208, the mapping function *f* is encoded in a computer readable medium, whereby the mapping function *f* is useful for generating mapping coordinates for additional products in the combinatorial library from building block features associated with the additional products. In one embodiment of the invention, the mapping function *f* is implemented in step 208 using a neural network. In other embodiments, the mapping function *f* is implemented in step 208 using a set of specialized mapping functions f_1 through f_n. In some embodiments, each such specialized mapping function is implemented using a neural network. Other methods can also be used to implement the mapping function *f*. In embodiments of the invention, step 208 is performed in conjunction with step 206.

[0063] It accordance with the invention, the encoded mapping function *f* may be distributed and used by individuals to analyze virtual combinatorial libraries. In embodiments, the encoded mapping function *f* is distributed as a part of a computer program product. These computer program products can be used to perform optional step 210.

[0064] In optional step 210, building blocks features for at least one additional product in the combinatorial library are provided to the mapping function *f*, wherein the mapping function *f* outputs mapping coordinates for the additional product. The mapping coordinates produced by the mapping function *f* can be used, for example, to analyze, search, or classify additional products in the combinatorial library. When performed, optional step 210 can be performed

by the same person or legal entity that performed steps 202-208, or by a different person or entity.

B. Example Method 300

[0065] FIGs. 3A and 3B illustrate a flowchart of the steps of a second method 300 for generating coordinates for products in a virtual combinatorial library according to an embodiment of the invention. As will become apparent from the description, method 300 includes a combinatorial network training phase (steps 302, 304, 306, 308, and 310) and an optional product mapping coordinate generation phase (steps 312 and 314). The steps of method 300 will now be described with reference to FIGs. 3A and 3B.

[0066] In step 302 of method 300, a subset of training products $\{p_i,\ i = 1,2,\ldots,k;\ p_i \in P\}$ is selected from a combinatorial library P.

[0067] The training subset of products $\{p_i,\ i = 1,2,\ldots,k;\ p_i \in P\}$ selected in step 302 can be chosen in any manner. For example, the training subset can be chosen randomly or non-randomly. In most cases, the composition of a particular training subset does not have a significant influence on the quality of a map as long as it is representative of the combinatorial library from which it is selected. Empirical evidence suggests that for moderately large combinatorial libraries ($\sim 10^5$ products), a training subset on the order of 1-3% is usually sufficient to train a combinatorial network according to the invention.

[0068] In step 304 of method 300, features of choice are computed for each reagent or building block in the combinatorial library P, $\{f_{ijk},\ i = 1,2,\ldots,r;\ j = 1,2,\ldots,r_i;\ k = 1, 2, \ldots, n_i\}$, where r is the number of variation sites in the combinatorial library, r_i is the number of building blocks at the i-th variation site, and n_i is the number of features used to characterize each building block at the i-th variation site. At least one feature is computed for each reagent or building block. Features computed for the reagents or building blocks at a particular variation site in the combinatorial library may not be the same as features computed for the building blocks at different variation sites in the combinatorial library. In embodiments of the invention, at least some of the

reagent or building block features represent latent variables derived from other reagent or building block data, such as principal components, principal factors, MDS coordinates, etc.

[0069] In step 306, the training subset of products selected in step 302 is mapped onto $\Re^m$ using a nonlinear mapping algorithm ($p_i \rightarrow y_i$, $i = 1,2,\ldots,k$, $y_i \in \Re^m$) and a function of choice for assigning relationships between products. This function takes as input a pair of products or data associated with a pair of products and returns a numerical value that represents a relationship (similarity, dissimilarity, or some other type of relationship) between the products.

[0070] In embodiments of the invention, the nonlinear mapping algorithm ($p_i \rightarrow y_i$, $i = 1,2,\ldots,k$, $y_i \in \Re^m$) is any conventional multidimensional scaling or nonlinear mapping algorithm. In other embodiments, the nonlinear mapping algorithm ($p_i \rightarrow y_i$, $i = 1,2,\ldots,k$, $y_i \in \Re^m$) comprises the following steps to determine each y_i: (1) placing the training subset of products on an m-dimensional map at some initial coordinates; (2) selecting a pair of products from the training subset of products having a known or assigned relationship; (3) revising the mapping coordinates of one or both of the selected products based on their assigned relationship and the corresponding distance between the products on the map so that the distance between the products on the m-dimensional map are more representative of the assigned relationship between the products; and (4) repeating steps (2) and (3) for additional pairs of products from the training subset of products until a stop criterion is satisfied. This mapping process is further described in U.S. Patent No. 6,295,514, and U.S. Application No. 09/073,845, filed May 7, 1998.

[0071] In other embodiments of the invention, the nonlinear mapping algorithm ($p_i \rightarrow y_i$, $i = 1,2,\ldots,k$, $y_i \in \Re^m$) comprises the following steps to determine each y_I: (1) placing the training subset of products on an m-dimensional map at some initial coordinates; (2) selecting at least three products having at least some known or assigned relationships; (3) revising mapping coordinates of at least some of the selected products so that at least some of the distances between the refined coordinates of at least some of the

selected products are more representative of corresponding relationships between the products; and (4) repeating steps (2) and (3) for additional subsets of products from the training subset of products until a stop criterion is satisfied. This mapping process is further described in U.S. Patent No. 6,295,514, and U.S. Application No. 09/073,845, filed May 7, 1998.

[0072] In step 308 of method 300, for each product p_i of the training subset of products, the corresponding reagents or building blocks $\{t_{ij}, j = 1, 2, \ldots, r\}$ of product p_i are identified and their features $f_{t_{i1}1}, f_{t_{i1}2}, \ldots, f_{t_{i1}n_1}, \ldots, f_{t_{ir}n_r}$ are concatenated into a single vector, x_i. A training set $T = \{(x_i, y_i), i = 1,2,\ldots,k\}$ is typically denoted.

[0073] In step 310, a combinatorial network is trained to reproduce the mapping $x_i \rightarrow y_i$ using the input/output pairs of the training set T. In embodiments of the invention, the combinatorial network and its associated parameters can be exported for use by other systems and/or computer program products.

[0074] Step 310 ends when the combinatorial network is trained. Once the network is trained, the network can be used to generate mapping coordinates for products of combinatorial library P in accordance with steps 312 and 314 of method 300.

[0075] In step 312, for each product $\{p_z, z = 1, 2, \ldots w\}$ of the combinatorial library P to be mapped onto $\Re^m$, corresponding reagents or building blocks $\{t_j, j = 1, 2, \ldots, r\}$ are identified, and their features $f_{t_{i1}1}, f_{t_{i1}2}, \ldots, f_{t_{i1}n_1}, \ldots, f_{t_{ir}n_r}$ are concatenated into a single vector, x_z. The features of step 312 are the features computed in step 304.

[0076] In step 314, the trained combinatorial network is used to map $x_z \rightarrow y_z$, wherein y_z represents mapping coordinates for product p_z.

[0077] In embodiments of the invention, the mapping coordinates produced by the combinatorial network are stored for subsequent retrieval and analysis. The mapping coordinates can be analyzed, for example, using conventional statistical and/or data mining techniques. The mapping coordinates of the products can also be used, for example, to generate a similarity plot of the products for viewing on a display screen. Other methods for analyzing the

mapping coordinates of the products will be known to a person skilled in the relevant art given the description herein.

4. Exemplary Applications of the Invention

[0078] In this section, two exemplary applications of the present invention are presented. Both of these applications illustrate the generation of 2-dimensional mapping coordinates for the products of a combinatorial library given a set of computed descriptors (properties) of the library products and a molecular similarity function evaluated on the basis of these descriptors. The objective was to map the products in the combinatorial library onto a 2-dimensional map in such a way that the Euclidean distances of the products on the 2-dimensional map approximated as closely as possible the corresponding dissimilarities of the respective products. Thus, the computed dissimilarities of the products were used as a measure of the relationships between the products. The two exemplary applications differ in the function that was used to measure the dissimilarity between two products in the virtual library.

[0079] FIG. 4 illustrates the reductive amination reaction scheme 400 that was used to generate the combinatorial library used in the exemplary applications. In accordance with reaction scheme 400, a virtual library of 90,000 products was generated using a set of 300 primary amines and 300 aldehydes. A set of 300 primary amines and 300 aldehydes (i.e., 600 reagents or building blocks) were selected from the Available Chemicals Directory (a database of commercially available reagents marketed by MDL Information Systems, Inc., 140 Catalina Street, San Leandro, CA 94577, which is incorporated by reference herein in its entirety) and used in accordance with reaction scheme 400 to generate a library of 90,000 products.

[0080] Each of the 600 reagents and 90,000 products was described by two sets of descriptors: (1) Kier-Hall topological indices (KH), and (2) ISIS keys (IK). The former is a collection of 117 molecular connectivity indices, kappa shape indices, subgraph counts, information-theoretic indices, Bonchev-Trinajstić indices, and topological state indices. The latter are 166-dimensional binary vectors, where each bit encodes the presence or absence of a particular

structural feature in the molecule. The bit assignment is based on the fragment dictionary used in the ISIS chemical database management system.

[0081] To eliminate redundancy in the data, the Kier-Hall (KH) descriptors for the reagents and products were independently normalized and decorrelated using principal component analysis (PCA). This process resulted in an orthogonal set of 24 and 23 latent variables for the reagents and products, respectively, which accounted for 99% of the total variance in the respective data. To simplify the input to the neural networks, PCA was also applied to the binary ISIS keys, resulting in 66 and 70 principal components for the reagents and products, respectively.

[0082] In the case of the KH descriptors, the dissimilarity between two products was measured using the Euclidean distance in the 23-dimensional space formed by the products' principal components. For the ISIS keys, the dissimilarity between two products was measured using the Tanimoto distance:

$$S = 1 - T$$

where T is the Tanimoto coefficient:

$$T = \frac{|AND(x,y)|}{|IOR(x,y)|}$$

and where x and y represent two binary encoded molecules, *AND* is the bitwise "and" operation (a bit in the result is set if both of the corresponding bits in the two operands are set), and *IOR* is the bitwise "inclusive or" operation (a bit in the result is set if either or both of the corresponding bits in the two operands are set).

[0083] In the exemplary applications described herein, the training set was determined by random sampling. Thus, the analysis consisted of the following steps. First, a set of descriptors were computed for each of the reagents that make up the virtual library. A random sample of 3,000 products (the training subset of products) from the virtual library was then identified, and mapped to

two dimensions using the pairwise refinement method described above. This method starts by assigning initial mapping coordinates to the training subset of products, and then repeatedly selects two products from the training subset of products and refines their coordinates on the map so that the distance between the coordinates on the map corresponds more closely to the relationship (dissimilarity) between the selected products. This process terminates when a stop criterion is satisfied. The resulting coordinates were used as input to a combinatorial network, which was trained to reproduce the mapping coordinates of the products in the training subset of products from the descriptors of their respective building blocks. Once trained, the neural network was used in a feed-forward manner to map the remaining products in the virtual library. For comparison, the map derived by applying the pairwise refinement process on the entire virtual library was also obtained. These reference maps are shown in FIG. 5A and 6A for the KH and IK descriptors, respectively.

[0084] The results discussed herein were obtained using three-layer, fully connected neural networks according to the invention. The neural networks were trained using a standard error back-propagation algorithm (see, e.g., S. Haykin, Neural Networks, Macmillan, New York (1994)). The logistic transfer function $f(x)=1/(1+e^{-x})$ was used for both hidden and output layers. Each network had 10 hidden units and was trained for 500 epochs, using a linearly decreasing learning rate from 1.0 to 0.01 and a momentum of 0.8. During each epoch, the training patterns were presented to the network in a randomized order.

[0085] For the KH maps, the input to the neural network consisted of the reagent principal components that accounted for 99% of the total variance in the reagent KH descriptors. For the IK maps, the input to the neural network consisted of the reagent principal components that accounted for 99% of the total variance in the reagent IK binary descriptors.

[0086] The maps obtained with the combinatorial networks trained using the aforementioned procedure are illustrated in FIG. 5B and 6B for the KH and IK descriptors, respectively. As illustrated in FIGs. 5A-B and 6A-B, in both

cases, combinatorial networks trained according to the invention produced maps that were comparable to those derived by fully enumerating the entire combinatorial library (FIG.5A and 6A, respectively). A more detailed study of the effects of network topology and training parameters, sample size, sample composition, structure representation, input and output dimensionality, and combinatorial complexity is described in (Agrafiotis, D. K., and Lobanov, V. S., Multidimensional Scaling of Combinatorial Libraries without Explicit Enumeration, *J. Comput. Chem.*, 22, 1712-1722 (2001)), which is incorporated herein by reference in its entirety.

[0087] Although the preceding examples focus on 2-dimensional projections, the invention can also be used for mapping products into higher dimensions in order to facilitate their analysis by established statistical methods. Martin *et al*, for example, has used this technique to convert binary molecular fingerprints into Cartesian vectors so that they could be used for reagent selection using D-optimal experimental design (See, e.g., Martin, E., and Wong, A., Sensitivity analysis and other improvements to tailored combinatorial library design. J. Chem. Info. Comput. Sci., *40*, 215-220 (2000), which is incorporated by reference herein in its entirety).

5. System and Computer Program Product Embodiments

[0088] As will be understood by a person skilled in the relevant arts given the description herein, the method embodiments of the invention described above can be implemented as a system and/or a computer program product. FIG. 7 shows an example computer system 700 that supports implementation of the present invention. The present invention may be implemented using hardware, software, firmware, or a combination thereof. It may be implemented in a computer system or other processing system. The computer system 700 includes one or more processors, such as processor 704. The processor 704 is connected to a communication infrastructure 706 (e.g., a bus or network). Various software embodiments can be described in terms of this exemplary computer system. After reading this description, it will become apparent to a

person skilled in the relevant art how to implement the invention using other computer systems and/or computer architectures.

[0089] Computer system 700 also includes a main memory 708, preferably random access memory (RAM), and may also include a secondary memory 710. The secondary memory 710 may include, for example, a hard disk drive 712 and/or a removable storage drive 714, representing a floppy disk drive, a magnetic tape drive, an optical disk drive, etc. The removable storage drive 714 reads from and/or writes to a removable storage unit 718 in a well-known manner. Removable storage unit 718 represents a floppy disk, magnetic tape, optical disk, etc. As will be appreciated, the removable storage unit 718 includes a computer usable storage medium having stored therein computer software and/or data. In an embodiment of the invention, removable storage unit 718 can contain input data to be projected.

[0090] Secondary memory 710 can also include other similar means for allowing computer programs or input data to be loaded into computer system 700. Such means may include, for example, a removable storage unit 722 and an interface 720. Examples of such may include a program cartridge and cartridge interface (such as that found in video game devices), a removable memory chip (such as an EPROM, or PROM) and associated socket, and other removable storage units 722 and interfaces 720, which allow software and data to be transferred from the removable storage unit 722 to computer system 700.

[0091] Computer system 700 may also include a communications interface 724. Communications interface 724 allows software and data to be transferred between computer system 700 and external devices. Examples of communications interface 724 may include a modem, a network interface (such as an Ethernet card), a communications port, a PCMCIA slot and card, etc. Software and data transferred via communications interface 724 are in the form of signals 728 which may be electronic, electromagnetic, optical or other signals capable of being received by communications interface 724. These signals 728 are provided to communications interface 724 via a communications path (i.e., channel) 726. This channel 726 carries signals 728 and may be implemented using wire or cable, fiber optics, a phone line, a cellular phone link, an RF link and other communications channels. In an

embodiment of the invention, signals 728 can include input data to be projected.

[0092] Computer programs (also called computer control logic) are stored in main memory 708 and/or secondary memory 710. Computer programs may also be received via communications interface 724. Such computer programs, when executed, enable the computer system 700 to perform the features of the present invention as discussed herein. In particular, the computer programs, when executed, enable the processor 704 to perform the features of the present invention. Accordingly, such computer programs represent controllers of the computer system 700.

6. Conclusion

[0093] While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example, and not limitation. It will be apparent to persons skilled in the relevant art that various changes in detail can be made therein without departing from the spirit and scope of the invention. Thus the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.